High End Computing Revitalization Task Force (HEC-RTF), Inter Agency Working Group (HEC-IWG) File Systems and I/O Research Guidance Workshop

Rob Ross DOE/Office of Science ANL
Evan Felix DOE/Office of Science PNL
Bill Loewe DOE/NNSA LLNL
Lee Ward DOE/NNSA SNL
James Nunez DOE/NNSA LANL
John Bent DOE/NNSA LANL
Gary Grider DOE/NNSA LANL
Ellen Salmon NASA
Marti Bancroft DOD/NRO

Executive Summary

The need for immense and rapidly increasing scale in scientific computation drives the need for rapidly increasing scale in storage for scientific processing. Individual storage devices are rapidly getting denser while their bandwidth is not growing at the same pace. In the past several years, initial research into highly scalable file systems, high level Input/Output (I/O) libraries, and I/O middleware was done to provide some solutions to the problems that arise from massively parallel storage. To help plan for the research needs in the area of File Systems and I/O, the inter-government-agency document titled “HPC File Systems and Scalable I/O: Suggested Research and Development Topics for the Fiscal 2005-2009 Time Frame” ftp://ftp.lanl.gov/public/ggrider/HEC-IWG-FS-IO-Workshop-08-15-2005/FileSystems-DTS-SIO-FY05-FY09-R&D-topics-final.pdf was published which led the High End Computing Interagency Working Group (HEC-IWG) to designate this area as a national focus area starting in FY06. To collect a broader set of research needs in this area, a workshop was held on August 16-17, 2005 in Grapevine TX. Government agencies, top universities in the I/O area, and commercial entities that fund file systems and I/O research were invited to help the HEC-IWG determine the most needed research topics within this area. The information gathered at this workshop will be used to facilitate better coordinated government funded research in this important area in the coming years. An advisory group will be formed to continue the file systems and I/O research coordination effort for the coming years for the HEC-IWG.

The workshop concentrated on file systems and I/O middleware, and only addressed areas like high level data management I/O libraries, archive, and Grid as they had an effect on file systems and I/O, since these other areas already have other venues to facilitate discussion about research. The workshop attendees helped:

- Catalog existing government funded, and other relevant, research in this area,
- List top research areas that need to be addressed in the coming years both short and long term,
- Determine where gaps and overlaps exist, and
- Recommend the most pressing future short and long term research areas and needs and other actions necessary to ensure a well coordinated set of government funded research in this area.

During the workshop, a number of research themes emerged. The recommended research topics are organized around these themes which include metadata, measurement and understanding, Quality of Service (QoS), security, next-generation I/O architectures, communication and protocols, and management and RAS.

Both evolutionary and revolutionary research into metadata issues is needed, especially in the areas of scalability, extensibility, access control, reliability, availability, and longevity for both file and archival systems. Additionally, consideration for very revolutionary ideas such as new approaches to name spaces and use of novel storage devices needs to be explored.

Research into measurement and understanding of end-to-end I/O performance is needed including evolutionary ideas such as layered performance measurement, benchmarking, tracing, and visualization of I/O related performance data. Also, more radical ideas like end-to-end modeling and simulation of I/O stacks and the use of virtual machines for large scale I/O simulation need to be explored.

QoS is a ripe topic for research especially in the area of providing prioritized, deterministic performance in the face of multiple, complex, parallel applications running concurrently with other non-parallel workloads. More revolutionary ideas such ad dynamically adaptive end-to-end QoS throughout the hardware and software I/O stack are equally important.

Aspects of security such as usability, long term key management, distributed authentication, and dealing with security overhead are all good topics for research. There is also room for more difficult research topics such as novel new approaches to file system security including novel encryption end-to-end or otherwise that can be managed easily over time. The need for standardization of access control list mechanisms is also needed and investigation into a standard API for end-to-end encryption could be useful.

There is great need for research into next-generation I/O architectures, including evolutionary concepts such as extending the POSIX I/O API standard to support archives in a more natural way, access awareness, and HEC/high concurrence. Studies into methods to deal with small, unaligned I/O and mixed-size I/O workloads as well as collaborative caching and impedance matching are also needed. Novel approaches to I/O and File Systems also need to be explored including redistribution of intelligence, adaptive and reconfigurable I/O stacks, user space file systems, data-aware file systems, and the use of novel storage devices.

In the area of file system related communications and protocols, evolutionary items such as exploitation of Remote Direct Memory Access (RDMA), Object Based Secure Disk (OBSD) extensions, Network File System Version 4 (NFSv4) extensions, and parallel Network File System (pNFS) proof-of-concept implementations as well as more revolutionary exploration of server to server communications are needed.

In the area of management, reliability and availability at scale, management scaling, continuous versioning, and power management are all needed research topics. Additionally, more revolutionary ideas like autonomics, use of virtual machines, and novel devices exploitation need to be explored.

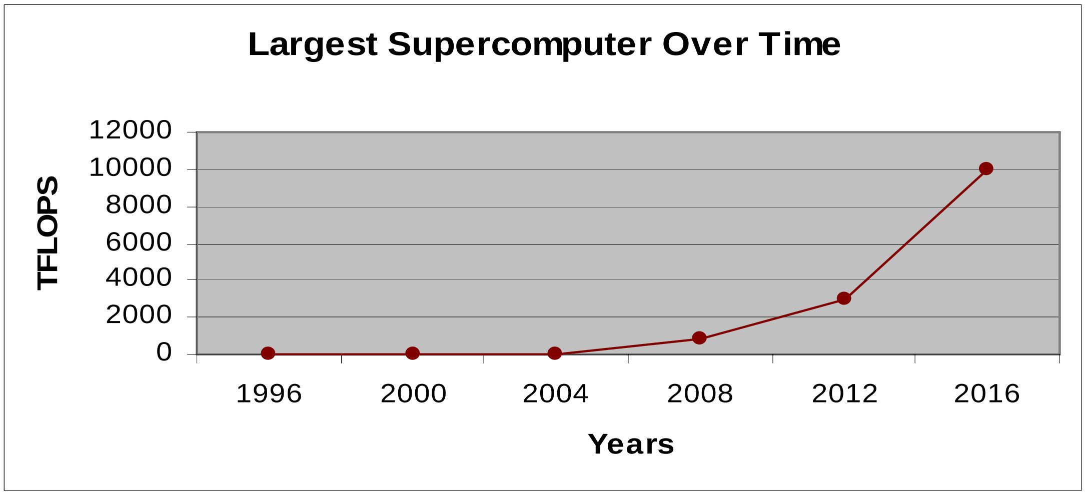
The purpose of this document is to present the areas of needed research in the HPC file systems and scalable I/O area which should be pursued by the government, that were identified at the HEC-IWG File Systems and I/O Research Guidance workshop

More focused and complete government research investment needs to be made in the file systems and I/O middleware area of HEC, given its importance and its lack of sufficient funding levels in the past, as compared to other elements of HEC. Scalable I/O is perhaps the most overlooked area of HEC R&D, and given the information generating and processing capabilities being installed and contemplated. It is a mistake to continue to neglect this area of HEC. Many areas in need of new and continued investment in R&D and standardization in this crucial HEC I/O area have been summarized in this document.

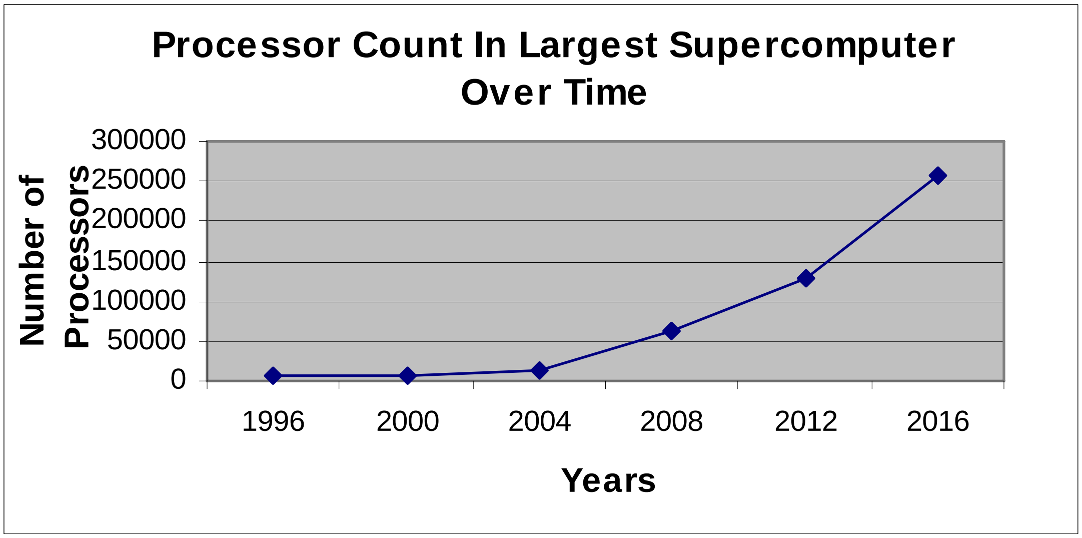
Compelling Case Information

This section of the document uses historical trends and projected future trends to explain why I/O and file system research is needed. It also uses these past and future projected trends to explain why the workshop attendees settled on the identified set of research topics, which can be equated to current and anticipated areas of concern for future I/O and file systems.

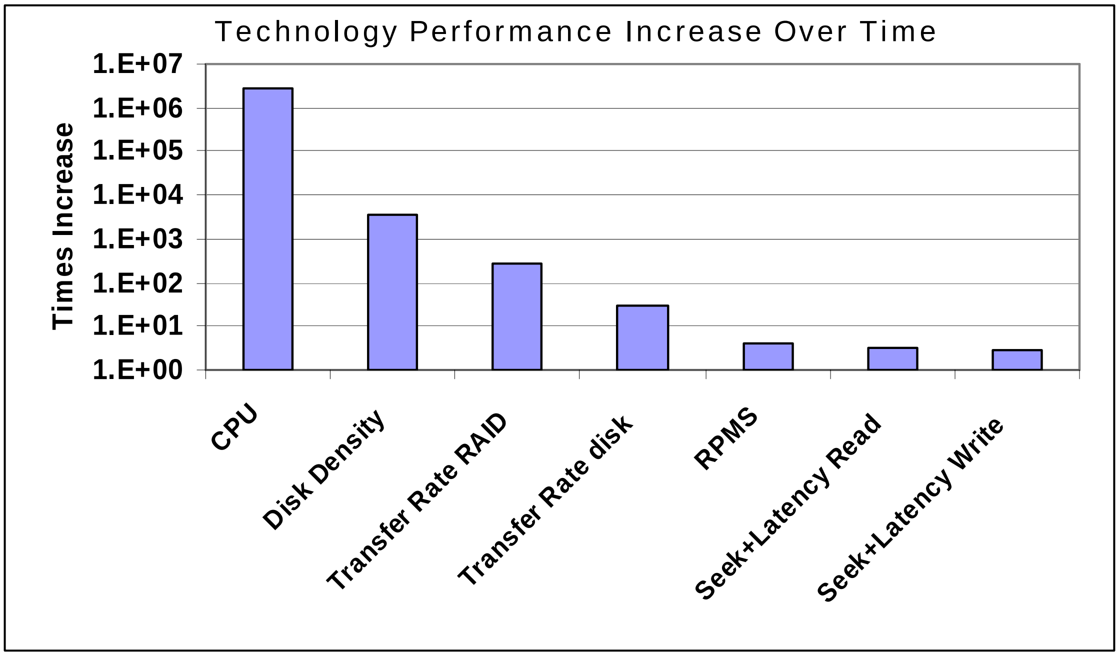
Although processor clock speeds have grown drastically over the last two decades, it appears that the rapid increases in processor clock rates are slowing. The microprocessor industry is exploring and even beginning to deploy processor architectures that have many more processing units per chip or board to continue to meet the processing power growth demand. This implies that scientific applications will have to begin to rely more heavily on multi-process/task parallelism at a greater scale than ever before. When single processors were getting much faster each year, applications could gain advantage over time by keeping a constant number of processes/tasks, but this appears to be no longer true. In order for applications to continue to gain speed up, it will now require the use of more processors over time. The following graph illustrates the past and future anticipated performance we have seen and expect to see on large scientific computing platforms.



The compute capabilities of machines anticipated for scientific computing is growing rapidly. The next graph illustrates the corresponding growth in the number of processors we expect to see used to build these large scientific computers.

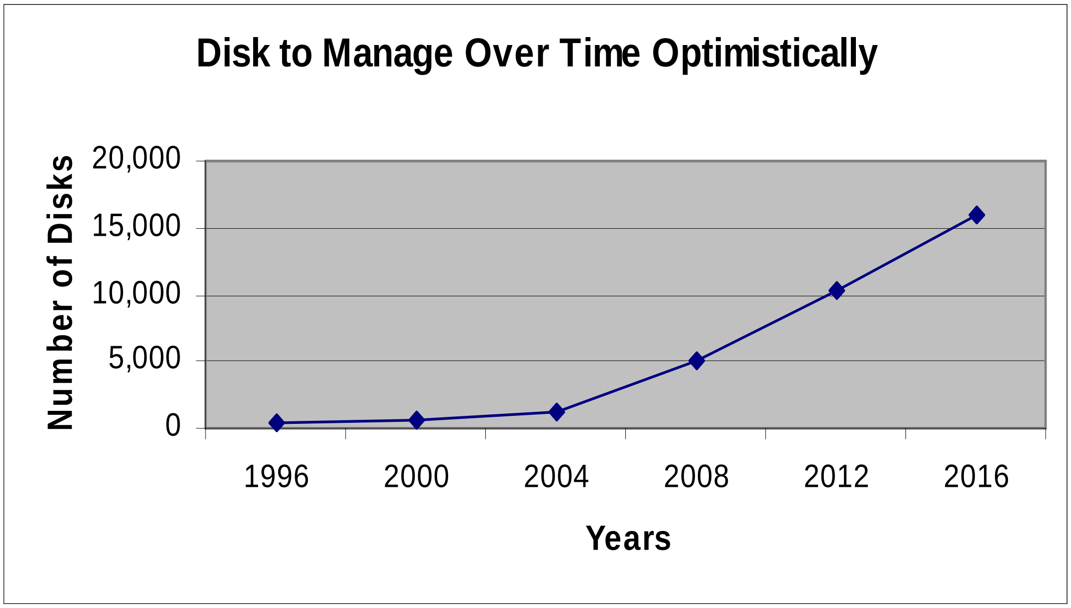


This following graph is a demonstration of the relative performance improvements from 1977 to 2005 for CPU performance as compared to disk storage performance related metrics. (courtesy Henry Newman, Instrumental Inc. for DARPA.)



Increases in processor speed and disk density have both grown at alarming rates while disk transfer rates have only grown modestly and disk agility has hardly improved at all.

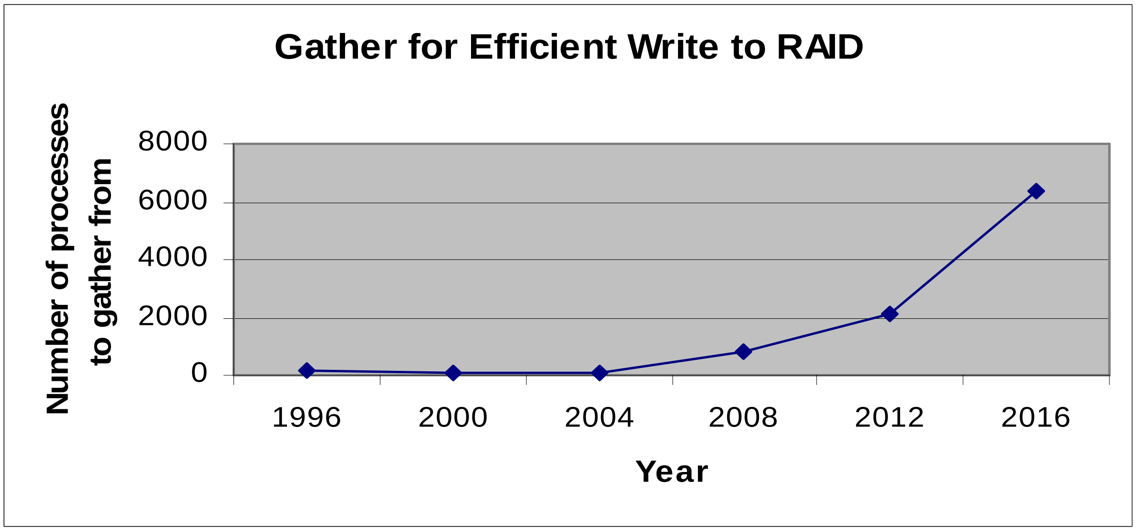
Observing the rate of growth in processing power in HEC systems and estimating the data rates necessary to avoid processor starvation (band width and latency) implies that the number of disks required to meet this demand will grow radically. The graph below shows a very optimistic view of the number of disk drives needed to provide I/O service for large scientific computers over time.



Another trend in extreme scale supercomputers is that the amount of affordable memory per processor is not growing at a substantial rate.

The combination of needing vastly more processors, constant or less memory per processor, and the dismal improvement in performance characteristics for disk storage devices, has the effect that the number of processes one has to gather from for efficient writes to the storage devices goes up radically.

The following diagram illustrates this phenomenon.



The need to gather from many more processors/memory in order to perform efficient I/O operations generates the need for handling much more in flight data to maintain performance.

These trends, including the presence of more processors with less memory per process, the need to deal with far more disk drives concurrently than ever before, and the need to deal with far more in flight data than ever before, form the basis for many of the issues faced in the I/O and file systems area of the HEC environment.

Two categories of issues that arise from the trends are:

1) How do you manage 100,000 mechanical disk drive devices and their associated environment, both hardware and software? This includes, RAS, QoS for usage that varies by 7 orders of magnitude, management without requiring an army of administrators, security, etc.

2) How do you productively use 100,000 mechanical disk drive devices and their associated environment? This includes middleware, high level libraries, file systems, dealing with massive in flight data, etc.

It is no surprise that the above trend information describes well most of the existing and new I/O and file systems issues identified at the workshop.

Despite these troubling trends in processor and storage, an examination of the quantity of computer science research in I/O and file systems compared to other areas of the HEC environment reveals that the I/O and file systems area has been greatly neglected in comparison. This helps explain why investments in research in this area are needed so acutely. The area of I/O has really been largely overlooked both in the area of how to manage enormous scale I/O systems and how to productively use such systems.

Background

The need for immense and rapidly increasing scale in scientific computation drives the need for rapidly increasing scale in storage for scientific processing. Individual storage devices are rapidly getting denser while bandwidth is not growing at the same pace.
In the past several years, initial research into highly scalable file systems, high level Input/Output (I/O) libraries, and I/O middleware was done to provide some solutions to the problems that arise from massively parallel storage. To help plan for the research needs in the area of File Systems and I/O, the inter-government-agency document "HPC File Systems and Scalable I/O: Suggested Research and Development Topics for the Fiscal 2005-2009 Time Frame" was published which led the HEC-IWG to designate this area as a national focus area starting in FY06. To collect a broader set of research needs in this area, the HEC-IWG File Systems and I/O Research Guidance Workshop was held on August 16-17, 2005 in Grapevine TX. HEC government agencies, top universities in the I/O area, and commercial entities that fund file systems and I/O research were invited to help the HEC-IWG determine the most needed research topics within this area. The information gathered at this workshop will be used to facilitate better coordinated government funded research in this important area in the coming years. An advisory group will be formed to continue the file systems and I/O research coordination effort for the coming years for the HEC-IWG.

The workshop concentrated on file systems and I/O middleware, and only addressed areas like high level data management I/O libraries, archive, and Grid as they had an effect on file systems and I/O, since these other areas already have other venues to facilitate discussion about research. The workshop attendees helped:

- Catalog existing government funded, and other relevant, research in this area,
- List top research areas that need to be addressed in the coming years both short and long term,
- Determine where gaps and overlaps exist, and
- Recommend the most pressing future short and long term research areas and needs and other actions necessary to ensure a well coordinated set of government funded research in this area.

Frequently used terms

I/O – input/output
File system – A combination of hardware and software that provides applications access to persistent storage through an application programming interface (API), normally the Portable Operating System Interface (POSIX) for I/O.
POSIX - Portable Operating System Interface (POSIX), the standard user interfaces in the UNIX based and other operating systems (http://web.access.net.au/felixadv/files/output/book/x1164.html)
Global – refers to accessible globally (by all), often implies all who access a given resource see the same view of the resource
Parallel – multiple coordinated instances, such as streams of data, computational elements

Scalable – decomposition of a set of work into an arbitrary number of elements, the ability to subdivide work into any number of parts from 1 to infinity
Metadata – information that describes stored data; examples are location, creation/access dates/times, sizes, security information, etc.
Higher level I/O library – software libraries that provide applications with high level abstractions of storage systems, higher level abstractions than parallelism, examples are the Hierarchical Data Formats version 5 library (HDF5) (http://www-rcd.cc.purdue.edu/~aai/HDF5/html/RM_H5Front.html) and parallel Network Common Data Formats library (PnetCDF) (http://www-unix.mcs.anl.gov/parallel-netcdf/sc03_present.pdf)
I/O Middleware – software that provide applications with higher level abstractions than simple strings of bytes, an example is the Message Passing Interface – I/O library (MPI-IO) (http://www-unix.mcs.anl.gov/romio)
WAN – Wide Area Network, refers to connection over a great distance, tens to thousands of miles
SAN – Storage Area Network, network for connecting computers to storage devices
QoS – Quality of Service

Research Themes Identified From the Workshop

During the workshop, a number of research themes emerged. The recommended research topics are organized around these themes: metadata, measurement and understanding, Quality of Service, security, next-generation I/O architectures, communication and protocols, and management and RAS. The following subsections describe the research appropriate to each theme area. Each subsection will cover both evolutionary and more revolutionary research topics that were identified as needing additional research attention.

Metadata

Evolution

A number of research directions target immediate needs in the area of metadata management.

Scalability
Clustered file systems seem to be converging on an architecture that employs a centralized metadata service to maintain layout and allocation information among multiple, distinct movers. While this has had significant, positive impact on the scalability in the data path, it has been at the expense of the scalability of the metadata service. The transaction rate against the metadata service has increased, as has the amount of information communicated between the metadata service component and its clients has increased. These trends indicate that distributed metadata storage is key.

Extensibility
HEC applications are increasingly creating, storing, and relying on derived data and provenance information as part of the discovery process. At the moment, additional databases or files are used to store this information. At the same time, extended attribute

support is becoming a common feature of local file systems. Additional work is needed to understand appropriate storage mechanisms for these user-created attributes within cluster and parallel file systems and to create interfaces to this data.

Metadata and Archiving

Archiving of cluster file systems is problematic for a number of reasons. One key factor is the vast number of individual items (e.g. files) that must be archived. Efficiently storing the metadata for these objects in a manner that is also efficiently accessed on streaming storage is an unsolved challenge.

Access Control Lists

Access Control Lists (ACLs) are a widely-implemented mechanism for limiting access to file system objects. The challenge in applying ACLs to cluster file systems is in their distributed nature. As we move away from centralized metadata storage to distributed metadata storage, efficiently verifying permissions will become more complicated and communication-intensive. Novel approaches to distributing ACLs and maintaining consistency of ACLs in a distributed environment are necessary to prevent these checks from becoming an artificial I/O bottleneck.

Data Transparency

Traditionally file systems have operated in terms of streams of bytes. However, today's file systems are accessed by numerous, often heterogeneous systems. In order to store data in a platform-independent manner, high-level libraries are used to convert data prior to storage on the file system. Augmenting file systems to understand basic data format semantics would allow these types of operations to be moved into the file system proper, allowing the file system to make decisions on the best format for physical storage and likely reducing the overhead of data recoding.

Revolution

Additional research is needed to address longer-term concerns in the area of metadata storage and management.

Scalability

In the near-term we are likely to see relatively simple metadata distribution schemes used to allow for more concurrency in metadata operations for cluster file systems with a moderate number of metadata storage devices. In the longer term it will be necessary to extend these schemes in order to facilitate the use of very large numbers of metadata storage devices. Techniques for discovery of file objects and mapping of tree-based name spaces onto in a very large metadata space are just two potential research areas for long-term study of metadata scalability. Continued scaling implies an increase in in-flight data and metadata, and adapting to this change is an area of great interest for scaling research as well.

Extensibility

As extendible metadata and data transparency become more common, it is likely that the file system will know increasingly more about the data being stored. With respect to metadata, an important issue will be how to store semantic information alongside file data in a matter that is accessible and understandable by the file system itself.

Name Spaces
In order to assist applications with managing enormous amounts of data, application programmers and data management specialists are calling for the ability to store and retrieve data in organizations other than the age-old file system tree-based directory structure. Data format libraries currently provide some of this function but are not at all well-mated to the underlying file system capabilities. Databases are often called upon to provide these capabilities but they are not designed for petabyte or exabyte scale stores with immense numbers of clients. Exploratory work in providing new metadata layouts and finding data in these new layouts is vital to address this identified need.

Metadata and Archiving
While simply archiving large volumes of data stored on cluster file systems is a challenge in itself, tighter coupling of storage system and archival system is desirable. One challenge in bringing file systems and archival systems together is the need for additional metadata describing locations of data, which might in fact not even exist on the file system at that point in time. Additional challenges to archiving of file system data come when name space changes occur. Capturing novel file system organizations on archival storage will require new approaches.

Hybrid Devices
New storage technologies such as MEMS, MRAM, FLASH, and others all provide storage that is faster than spinning disk, but at a higher cost. While it will be some time before such technologies supplant disk (if ever), these technologies are very amenable to use as metadata storage or metadata cache spaces. Integrating these devices into file system infrastructure holds the promise of increased metadata rates.

Measurement and Understanding

Research tools for measurement and understanding of parallel file system and end-to-end I/O performance are needed for advances in future file systems. In parallel application building and tuning, there are a multitude of correctness and performance tools available to applications. In the area of scalable I/O and file systems, however, there are few generally applicable tools available. Tools and benchmarks for use by application programmers, library developers, and file system managers would be an enormous aid.

There is a need for research into evolutionary ideas such as layered performance measurement, benchmarking, tracing, and visualization of I/O related performance data. More radical ideas to be explored include end to end modeling and simulation of I/O stacks and the use of virtual machines for large scale I/O simulation. With research into these ideas, a future generation of high performance file systems could be understood and more efficiently pursued.

Evolution

Understanding Layering Contribution

There needs to be research into layered performance measuring, primarily focused on understanding the contribution and effects in layering of the I/O stack. In this context, tools will need to be developed that can be integrated with other higher- and lower-level tools as a component of research into I/O stack layering. Such research would be critical in understanding file system design trade-offs in layering.

End-to-end Benchmarking and Tracing

As in other areas of computing, benchmarking is a vital part of the I/O professional's toolkit. Benchmarks in the areas of interactive benchmarks (simulating user experience items like ls, cp, rm, tar, etc.), throughput benchmarks (including both peak performance and I/O kernels from real scientific and engineering applications), and metadata benchmarks (collective opens, creates, resizes, etc) are needed. Some of these benchmarks exist and others do not. Further, new benchmarks will be necessary as new technologies are introduced into the file system – e.g., object-based benchmarks would be advantageous for testing object-based file systems. As a minimum, research in the benchmarking area is needed to collect and index current benchmarks and design and build new benchmarks to fill any gaps. A clearing house for all I/O benchmarks and their usage could be of significant benefit. It currently is quite difficult to determine if a benchmark exists for a particular function or workload.

Tools to quickly get tracing information from parallel applications that are representative of HEC I/O workloads, analyzing these traces to characterize the applications I/O footprint, and even being able to replay the traces in parallel against real or simulated parallel file systems would be of great use. Again, as with benchmarks, a tracing clearing house would be very valuable as a repository of known HEC I/O workloads. I/O tracing for applications would prove valuable in designing file systems for real world workloads.

Visualization

In the area of visualization, tools need to be designed and developed for the visualization of I/O-related performance. Currently, there exist few visualization tools which can make use of I/O performance data and traces for meaningful and intuitive analysis. The ability to have visualization tools provide trace-driven analysis for greater understanding would be significantly advantageous for I/O.

Revolution

End-to-end Modeling and Simulation

Tools for simulation of portions of the I/O stack or the entire global parallel file systems would also be of great use to assist in understanding design trade-offs for I/O performance for applications, libraries, and file systems. Most other areas of the computer industry rely heavily on simulation tools. While asking for a parallel file system simulator is a tall order, the value it would have could be enormous.

Virtual Machines as Tools for Large Scale Simulation

A further approach to I/O simulation would be the use of virtual machines for modeling and simulation of large scale I/O. This is an area which would likely prove to have great advantages for large scale I/O system understanding, research, and design.

QoS

Quality of Service can be defined as features of a storage architecture which allow a user or administrator to recommend policies for data movement during Input/Output operations. These Quality of Service policies can reach a broad range of integration into software, file systems, and hardware devices. Policies such as guaranteed I/O performance, specific redundancy requirements, or I/O priority settings will allow the system to perform optimally for a given work profile. Further research into areas such as adaptive QoS systems, end-to-end solutions, hardware support and cross-system integration will revolutionize storage systems that will be created in the next few years. These research topics will bring the storage systems to a point where users, systems, or entire clusters can be insulated from each other, while using the same storage infrastructure. This will also allow for predictable I/O performance and response time for the users.

Evolution

Guaranteed I/O Performance
Guaranteed I/O performance stretches the range of the I/O system. One user or process may need high-throughput I/O during a computation in order to use the processors more effectively, but want quicker meta-data responses while he is using the command line. This area is similar to how network QoS systems are defined, which provide better service for specific flows over the network, without stopping other flows. Storage systems bring a new face to QoS, as there are various types of flows needed, such as parallel I/O, meta-data response times, and redundancy needs.

Specific Redundancy Requirements
History has taught us that hardware fails, software fails, and sometimes users fail, consequently research into redundancy has been performed and technologies developed. Storage systems need to be able to use these abilities to their full advantage when needed and not use them when they are impeding the use of the system. Users will be storing many types of data on a storage system, from scratch data that does not need protection from failure, to data that absolutely never should be lost. Research into mechanisms that allow a user, system administrator, or policy maker to select the level of redundancy needed at run-time, and systems that will honor this selection, is necessary.

I/O Priorities
I/O priorities are also needed to fully implement a QoS system. They are different than guarantees of performance, as they are more of a meta-QoS system wherein the priority of one set of data can override other settings within the system. Things such as restoring redundancy requirements to a system with a failed component may be considered higher

priority than performance. Other things such as migration of data to archival systems may be set a lower priority than interactive I/O performance.

Revolution

Adaptive QoS

Once systems gain the ability to provide up-to-date information on how all the components of a system are performing, and on failure conditions, research into adaptive QoS can begin. A system that can recognize that a certain piece of hardware is slower than newer, faster components can pro-actively route high-throughput requests to the faster components. Adaptive systems should also be able to recognize potential failure conditions, and upgrade redundancy on possibly affected data. Other research into adaptivity could focus on recognizing performance characteristics of networks or paths to data and modifying the use of these resources as they evolve over time.

End-To-End QoS

The setting of policies or rules for how the system should react will be a great challenge. The need for complete component understanding of the QoS settings are will be essential to building complete QoS systems. The need for standards of how settings are communicated, and how they will be set and managed is a great research topic. These standards will also need to be combined with meta-data operations, management tools, security and protocols.

Hardware Support

Hardware systems will need to start supporting low-level QoS primitives to fully enable a complete system to use QoS effectively. As hardware components become more and more intelligent, they should be able to recognize requests and supply prioritized response times, or quicker access to data.

Cross-System Integration

As more computing centers move towards a single unified storage system that spans clusters, workstations, tape subsystems, and multiple sites, it will become necessary to have the system recognize that I/O streams will have signatures of some type and need to respond to those signatures according to the QoS policies they have been given. These signatures may reflect user supplied meta-data, redundancy requirements, cluster identification, or site identification. The ability of a system that uses a wide range of vendor hardware, network, and software packages to interoperate together in a cohesive system needs to be researched and developed.

Security

Security in file systems and I/O was a recurring theme at the workshop. A variety of security related topics were identified as needing research attention. The bulk of the issues identified were related to security function, usability, and overhead.

Evolution

Usability

The most popular security-related theme at the workshop had to do with usability of security. It was recognized that if security is not easy to both use and understand, people will not use it. Ease-of-use is one area of research for file system and I/O researchers to study. Current API's and user-level interfaces are just not easy enough to use to be adopted by the masses. Additionally, the current security interfaces make security capabilities hard to understand. It should be simple and obvious how to accomplish a particular security act in order for the overwhelming number of users of file systems and I/O systems to use security features. Once easy to use and understand API's and interfaces emerge, it will be important to standardize and validate them as well.

Key Management

Another recurring theme in the security area at the workshop was long-term key management. Security themes that incorporate encryption of data at rest require carefully thought out long-term key management schemes. While there are some products available in this area, none of these solutions have taken off in industry. There is still room for evolutionary research in the area of long-term key management. Issues like

1. What do you do when the encryption algorithm that encrypted the data at rest becomes easily breakable?
2. How do you protect the keys but allow for flexible use?
3. How do you manage the longevity of the keys over long periods of time and throughout natural events?

among others need to be addressed in ways that are compelling enough to be widely deployed in products.

Distributed Authentication and Authorization

Security systems must be able to handle authentication and authorization issues in a completely distributed world. Within a single HEC site there may be hundreds of thousands of entities that need to be protected. Additionally, many HEC endeavors require collaboration between multiple sites and organizations. Security solutions must provide ways of flexibly handling these situations as well. File systems and I/O security solutions must be able to participate in virtual organizations. There is room for research in the area of distributed security for file systems and I/O to address scale, distance, and flexibility.

Security Overhead

Security within file systems and I/O does not come for free. There will always be overheads associated with providing security. Given the massive scale that HEC environments represent, as well as geographically dispersed HEC sites, building security solutions that have acceptable overheads is a difficult task. There is a need for research into security overhead for HEC security applications.

Revolution

There is also a need for more revolutionary research to be done in the area of file systems and I/O security. Accomplishing flexible but yet secure systems and flexible and secure data at rest is a difficult topic. Making security simple to use and understand is also an area that could use some breakthrough thinking.

End- to-End Encryption API
The security of stored data is becoming increasingly important to industry and the government. There is clear benefit, then, for the provision of a robust, accepted application program interface that enables all data on the storage service and flying on the network to reside and move in encrypted form. Such an interface would provide for the encryption and decryption of data within the client-side operating system. It would also include definition of, or qualified reference to, multiple key management schemes such as public key interfaces and derived authentication (Kerberos/GSS-API). Without such an end-to-end API, the data security approach has to be enforced at each step, with consequent performance penalties and a reluctance to utilize encryption in HEC environments.

Next-Generation I/O Architectures

The predicted scale of future high-end computing provides the opportunity for a renewed examination of the I/O stack and architecture. For some time, developers have been working around legacy issues with existing systems and interfaces in order to provide IO bandwidth to applications. While these work-arounds are fairly well understood, they are increasingly awkward and many enforce stringent constraints. Often, these constraints are artificial. A fresh look at the IO stack and architecture may provide the ability to deal with typical high-end access patterns in a more natural way.

The I/O paradigm still in use traces back almost three decades, and it is based on the classic single-host-to-local-storage-device model (also described as an initiator-target model). Despite the development of storage devices with intelligence, and despite advances in the protocols used to transport data between host and storage device, development of shared file systems, object-oriented storage architectures and virtualization techniques, the underlying paradigm for direct access to storage devices remains one involving an initiator and a target. That paradigm imposes some severe restrictions in order to ensure data integrity. However, there is little incentive for vendors to make major changes in the paradigm given that many new acquisitions require interoperability with existing or at most next-generation computers and storage.

Discussions at the workshop identified some evolutionary areas needing a shorter-term focus as well as the opportunity for original research that could fundamentally change the way I/O to storage is performed. Because I/O is currently or rapidly becoming one of the limiting factors in HEC scaling for certain applications, this basic research is needed now.

Evolution

POSIX

The POSIX file access calls, simply, were not designed for high-end computing outside of an explicitly shared memory model. The environment for which it was designed assumed that file descriptors, synchronization, and buffer management were all supported in local, directly accessed, memory by very low-latency operations. In today's high-end computing, the dominant solution is a cluster or multi-programmed parallel machine, and these architectures directly expose a distributed memory. Instead of the current situation (vendors receiving exemptions from the POSIX standard, and acquisitions therefore having to be non-POSIX compliant for performance reasons), we need to move to a POSIX standard set that supports typical HEC file systems and file systems I/O access patterns.

Archive Considerations

The directed graph mandated by the POSIX name space extends well into hierarchical storage systems. However, while solutions such as X/DSM allow a useful transition, for the user, from one part of the storage hierarchy to another, IO access to the file address space is problematic. Users unexpectedly encounter long delays while data is copied to or from the high-overhead portions of the storage hierarchy, for instance.

The presence of near-line and off-line storage encourages the user to view the storage system as infinite. This is in direct conflict with site policies normally. Inevitably, administrators require a gate or hurdle the user must encounter so that it is understood that scratch, ephemeral, and redundant data should not be placed into the deeper, or archival, layers of the hierarchy. In the past, this has been accomplished by adding manual steps to the process or imposing quotas. Both of those methods, however, are counterproductive in a name space that should, or could, span multiple layers or the storage hierarchy. This will become even more of a concern as the name space is expected to be global within an enterprise.

Many sites have archival mandates where some data lives forever. For these sites, moving from one product to another is difficult as a migration of this data from the old product to the new must be performed. Then, too, the amount of this data is forever growing which, as time goes by, makes the problem ever more difficult. Even the identification of such data sets is not incorporated. While migration is not addressed at all, industry does address mandated archival needs by providing special write-once file system and hardware solutions. This is unnatural, though, as the partitioning of the name space is necessarily tied to policy. A more natural solution would allow policy to be applied to storage without regard to file location in the name space.

Access-Aware Interfaces

The POSIX standard mandates a "stream of bytes" view of the file address space. In high-end computing, however, many applications more naturally desire a view where the file address space is accessed in a disjoint fashion, from many sources simultaneously. Similarly, many applications make many disjoint accesses from a single source. While the application would view the entire transfer as a whole, POSIX forces it to be broken into many, smaller, transfers. An interface that would allow the application to specify the entire transfer in a single call would be useful.

HEC Considerations

POSIX, as has already been discussed, is not the most natural design for high-end IO. Alterations to the existing API or semantic changes might be beneficial. For instance, a collective open could mitigate considerable startup times on very large clusters.

Small/Unaligned I/O

Modern operating systems inevitably buffer IO transfers. While this has proved optimal in performance for locally attached storage it presents problems when the goal is to efficiently use remote storage that is byte granular. The client operating system will typically employ buffers of fixed size. An application that does not fill a buffer when writing, then, can place the operating system in the uncomfortable position of having to read a full buffer, update the content and then prematurely flush the modified content back to the stable store. If a buffer system was available that was variable length or naturally supported modified sub-regions then byte-granular stable stores such as object-based disk could be efficiently leveraged.

Another, related, goal is to minimize the number of buffering steps required for data. It is not always done in current software stacks, and that is a concern, both from a performance and a memory usage perspective.

Mixed Large and Small I/O

File system designs tend to require tuning to efficiently support either large or small transfers. Unfortunately, they do not seem amenable to supporting both simultaneously. Worse, some applications attempt both during different phases of their processing. Something adaptive is clearly called for, yet must be able to avoid performance penalties both for the bulk streaming I/O (where bandwidth dominates) and for the smaller transactions that are latency sensitive. Modern self-describing data organizations such as are employed by HDF and CDF/netCDF are able to scatter attributes throughout the file, interposed with the data. It is insufficient then to simply handle small and large transfers. We must also be able to handle these in a directed, or vectored, fashion. It would also be desirable to parse data format representation standards to the point where data sets could be reliably interchanged between heterogeneous platforms (processors, architectures, operation systems) by relying on libraries or other constructs to correctly interpret the data format API.

Collaborative Caching

A file’s address space is unqualified even when accessed by the several clients in a cluster or MPP machine. While this is an accepted, well understood paradigm, it amounts to globally shared memory without any hardware assists – Something long-ago recognized as sub-optimal.

Altering the interface would probably not be well received by the application developers, so some other approach must be taken. Joy's law tells us that we should not try to get rid of existing interface.

The core problem appears to be that a globally coherent view must be maintained. Many high-performance applications do not require such a thing. However, the service section in a high-end machine would.

The metadata service relies upon lock services to accomplish the globally coherent views. By definition, such a thing is provided by the cluster, or MPP, service section. Usually, these are a magnitude, or more, smaller in size than the compute client section as they are simple overhead – They enable the compute section but do not directly contribute. Worse, the available lock algorithms do not seem to scale, so even if a larger service section was available it would consume itself while trying to manage locks. Clearly, a renewed interest in scalable lock services is needed. Solutions involving the compute partition, to augment the power of the metadata service and relaxing the API semantics with respect to coherency could lessen the load on the metadata service section.

Impedance Matching

The existing IO software stack is deep and composed of different, sometimes disparate, modules. For instance, any distributed file system will rely on networking components. A fresh look at this stack, end-to-end, could address bottlenecks, especially when disparate modules call on each other. At the least, it would be highly desirable to have a standard method indicate a long latency path was in use so that normal timeouts were not employed. This is a problem with today's hierarchical storage management systems when part of the path is over a WAN. Object based storage systems will have a similar concern since their logical evolution is to have objects appear "equal" regardless of physical constraints (unequal file system behavior as well as distance).

Revolution

Redistribution of Intelligence and Rethinking the I/O Stack

In the HEC space, we have clearly outgrown the decades old initiator-target I/O paradigm, yet the requirements for performance, data integrity, and coherency remain. Achieving a revolutionary approach to I/O is constrained within the respected paradigm.

File system solutions in the high end have relied on a core stack from commodity file systems design for workstations and servers. This could be redesigned in order to add to, remove from, or alter the placement of existing file systems components in the software stack. For instance, a local buffer cache could be removed in favor of a collaborative cache maintained by a distributed application for its own use. Early research indicates a benefit here; however the only implementation has been in the presence of the local host buffer cache. Other components could be reexamined, in order to find potentially better placement within the stack.

Active disk, the ability to move part of an application near and on to the disk, has been an active area of research. However, no good interface and set of semantic rules has come along that would make it generally useful. Currently, it would seem that all solutions in this arena are restricted to modifications to support specific applications. A design that provides a "sandbox" so that multiple, unique applications can leverage the promise of active disk simultaneously would be welcome.

Adaptive/Reconfigurable Stack (application specific perhaps)

Many large engineering and physics simulations are burdened by CPU data cache coherency semantics. It's always been known that the ability to turn these off, where possible, enhances observed performance. Similarly, in the IO world, a distributed application does not require the services of a local buffer cache, often. Changes to the file system could be made that would react to or enable an application to remove these high-overhead but low value components from the control or data path could be usefully leveraged.

New Approaches

Existing solutions utilize the network as a communications channel but there is power in the network far beyond that. While some solutions go so far as to generate multiple, simultaneous, transfers there is not much that is fundamentally different from the classic initiator-target model. Typically, such things are done for simple efficiency reasons but ignore the real power in networks. Fresh approaches, such as peer-to-peer solutions use new paradigms to reorganize storage. Such solutions would directly incorporate geographical distance in their cost function and significantly lower the bar that prevents massive replication, enhancing fault characteristics, or directly leverage other attractive properties in the network. Truly novel thinking, such as the peer-to-peer solutions, would be attractive.

User-Space Component Considerations

While many research file systems utilize components in user space, the practice is uncommon for production file systems. Performance data in the high end, at least, would seem to suggest that this is a practical approach in general. A general approach to support such file systems is desirable. Benefits from the eased development and debugging effort might, conceivably, offset the slightly higher call latencies. One of the historic reasons for preventing user space file system activity has been the data integrity concern – research into mechanisms (shared secrets, others) that could allay those concerns and permit user space and system space file system behaviors to co-exist is desirable.

Semantically-Aware File Systems

File systems move bytes. Some government agencies believe that a file system augmented with knowledge of the data stored and transported could go much further. For instance, a file system that understood the machine word format used on the machine where the data was originally deposited could reformat for a heterogeneous network of machines when required. As well, understanding the relationships between records accessed by an application could allow the file system to do a better job when storing the data or use much more intelligent prefetch strategies when retrieving it. Providing a method for the definition of such associations could be useful. Then, researching how changes within the file system might leverage such information would be appropriate. Long-lived data will need to convey format many years into the future, potentially. The concern is not just word lengths and endian issues, but such things as floating point formats (mostly resolved) and the representations of complex math components. A generic API that could stand the test of time is desirable. Efficiency, in performance, CPU cycle consumption, and storage will remain an issue.

Novel Devices/Hybrid Devices Exploitation

Storage devices have not changed fundamentally in 50 years. The advancements possible in allocation policy and layout given a radical change in the access latency to bandwidth ratios seem attractive to explore. New devices with promise along these lines as well as hybrids combining existing storage solutions could spark renewed interest in, and value from, these core file system areas.

The focus, today, is on capacity. Obviously, there is a need. However, the increasing capacities also come at a price. The greater probability of faults on a single unit now jeopardizes RAID systems at rebuild time. One avenue that is being explored in depth is to use the aggregate to offset the error probabilities. Others could be in the individual disk units themselves.

Communications and Protocols

The increasing use of parallel file systems places a great burden on vendors to support a diverse and rich collection of clients. Most of the tasks accomplished by any such file system are similar, though. An active research program into generating common, well accepted APIs and protocols, then, appears worthwhile. Such research would strive to support the most common tasks while retaining an ability to be easily extended so that competition via unique features and capabilities is retained.

Evolution

Exploitation of Advanced Network Adapters

The emerging ability of commodity adapters to offload and even entirely manage network data transfers has sparked efforts to modify IO data paths to leverage this capability. Currently there are no common, well accepted, interfaces for accessing these capabilities. An interesting area of study involves the development of such interfaces, including support for RDMA capabilities and taking into account that not all HEC systems can easily support implementations involving interrupts.

Object-Based Storage Devices

The benefit of object-based storage is clear. While everyone seems to agree that the work is not finished, it is not clear at this time what new capabilities might be desired in these devices or how those capabilities might be leveraged by file systems or middleware.

Advances in this area could have a broad impact on metadata storage and file system organization.

HEC Extensions to NFSv4

NFS version 4 shows great promise for supplanting the version 3 implementations. This is especially attractive to end-users subject to laws and regulations governing access, use, retention, and dissemination constraints because of the NFS v4 ACL and security mechanisms. For HEC, however, the immediate benefits are not clear. More research into HEC extensions to NFS v4 might uncover additional opportunities for improving performance, such as leveraging RDMA for transfers. These extensions could be implemented through the NFS v4 minor versioning support.

pNFS Advances

As mentioned in the previous section, the NFS v4 minor version provisions allow one to extend the protocol. Recently, work has begun to extend it so that multiple, simultaneous network transfers may move data to and from the client in parallel. These early prototypes have shown promise, but other design points have yet to be explored. Additionally, only preliminary work has been performed in how to best connect parallel file systems with NFS v4 and the pNFS extensions. Additional research in combining parallel file systems and NFS v4, with an emphasis on coherency, consistency semantics, and preserving direct I/O paths is warranted.

Revolution

Server-to-Server Communication

The design and implementation of parallel file systems often requires server to server communications in order to manage coherency and locking, for instance. Many of the tasks accomplished by any such file system in the “back-end”, then, are common. At this time all such communication is performed using proprietary protocols. Designing a common protocol, or set of protocols, for server-to-server communication could provide greater interoperability and thus create more options for HEC system configuration. Such a design would naturally need to consider the possible need for extensions and the right level of abstraction at which to operate.

Management and RAS

Management and RAS (Reliability, Availability, and Serviceability) are both obvious areas affected by immense scale. The number of storage devices, associated hardware, and software needed to provide the needed scalable file system service in a demanding and mixed workload environment of the future will be extremely difficult to manage given current technology. Advances must be made in massive scale storage management to enable management survival with future file system deployments. Additionally RAS at scale is another major issue. Given that future file systems will be based on tens of thousands or more mechanical devices with an extremely complicated software stack deployed at scale, it is likely that failure will be more then norm than the exception.

Evolution

End to End RAS at Scale and its Associated Overhead

To provide the needed file system bandwidth and I/O operations to future clusters, unprecedented numbers of storage devices will need to be used in a coordinated way. Striping data from a single application over enormous numbers of disks will eventually lead to difficulties in protecting against data unavailability or loss of integrity. Current RAID protection and availability technologies are not designed to provide sufficient protection for such immense scale. Additionally, RAS must be addressed from an end-to-end point of view, so availability involves not only disk drives but also all the associated networks, hardware and software. Items like server failover, client process migration, End-to-end RAS and integrity at the 100,000 storage device scale with acceptable overhead is simply not an understood problem.

End-to-End Management at Scale

Another area affected by immense scale is management. The number of devices needed to provide the needed scalable file system service in a demanding and mixed workload environment of the future will be extremely difficult to manage given current technology and the complex software stack and associated hardware. Advances must be made in massive scale storage management to enable management survival with future file system deployments. This is another area that is not well understood at the scale of 100,000 storage devices.

Continuous Versioning

In the area of metadata, volume, object, and file management, research into the concept of continuous versioning might provide another dimension to assist in the management of volume, object, and file management and could also assist in the area of interfacing file systems with archives and backup systems.

Power Management

Just as power management at scale is a problem for the computational resource, it is also a very big problem for the storage resource as well. Storage however, presents an interesting problem in that mechanical devices are in use in storage systems. At immense scale, power management of storage devices is also not an understood problem.

End-to-End Performance Management

Due to the complexity of immense scale file storage systems, there are many aspects of performance management that are not understood problems. These problems start with just configuration of such a system and only get worse when trying to tune a storage system for usage that spans seven orders of magnitude in performance characteristics. Providing deterministic performance for this wide variety of usage with multiple workloads occurring simultaneously is a very hard problem. Additionally, providing deterministic performance in the face of frequent failure at scale is also a challenging problem. Of course, as in RAS, performance management is an end-to-end problem, involving networks, storage hardware, and software, making this area even more difficult to tackle. Dynamic feedback systems, modeling, and appropriate information gathering at scale with acceptable overheads are all possible ways to begin to tackle this problem. This item is a world class problem and in definite need of research.

Revolution

Autonomic Storage Management Concepts
The storage industry is currently working on management solutions that are fully automated. Ideas like, storage that self configures, self heals, self migrates, and self tunes are all being pursued. These ideas are all good ideas but none of these solutions are understood at the 100,000 storage device scale. Additionally if these features are to be useful in the HEC environment, where things like determinism in parallel are important, it would be very useful for the HEC community to be involved in this Autonomic Storage Management research to ensure that these good ideas are implemented in a way that the HEC community can benefit. For over a decade, cluster management tools have been being researched and developed with millions of dollars having been invested. Only now after that decade of R&D is the HEC community understanding how to manage 100,000 processing units. Autonomics is a very new topic and research into this area is in its infancy. This is an area where the HEC community is really behind the curve.

Virtual Machines and/or Novel Devices Management and RAS Enablers
The management and RAS problems for storage systems at scale are such formidable problems, thinking out of the box should be done. Simulation and modeling may be of great value in this area. Use of virtual machines to simulate scale as well as file system and storage device simulation may also be helpful. Additionally application of novel devices in both management and RAS may be worth pursuing.

Archive

This workshop was not about archive R&D, but since archives interact with I/O and file systems, this section concentrates on those interactions.

High-performance archives remain an integral part of balanced HEC systems. Agencies continue to fund research in many problems of particular interest to archives (e.g., content-related metadata, knowledge management, data discovery), but as noted above, high-performance archives and file systems depend on many of the same underpinnings. Thus advances in many of these will also improve archives' ability to serve HEC systems.

But the vast scale and longevity of data in HPC archives adds some particular slants to these research areas:

- Higher densities and lower prices for disk have begun to make disk archives feasible, but the complicating factors of RAS at the very large scale are a foremost concern for deep archives on disk.

- Long-lived archives experience extremes in namespace size, making efficient storage, management, and retrieval of file system metadata imperative, and research into new namespace technologies attractive. Content-addressable storage and similar technologies show promise in finding, tracking, and managing large archives over long periods, but more work is needed.
- The longevity of archive files makes more imperative the ability to set and enforce policy to manage the data. Policy is also important in the lifecycle movement of data among layers in the archive hierarchy.
- The X/DSM POSIX file system interface for offline data is more than ten years old; modern, high-performance archives call for a new generation replacement.
- Long-term archives must contend with migration to new generations of hardware; emerging technologies such as object-based storage architectures may be particularly well suited for optimizing such movement, and for enabling larger scale parallelism in the archive system.

Assisting Research

One frequently echoed problem expressed at the workshop was the difficulty faced by many researchers of working in the area of HEC. This problem seemed to particularly affect academic researchers who lack access to real HEC applications and computing platforms. As such, government investment is needed to support efforts that allow these researchers access to both the physical and virtual infrastructure that they need in order to participate in HEC research. Additional government investment is needed to support the growth of students working on I/O research within the HEC area.

Physical infrastructure (testbeds)

Many would-be HEC researchers, particularly those in academics, lack access to the large parallel computers typically used in HEC applications. Therefore, the development and maintenance of open testbeds would enable these researchers to contribute in the area of HEC. There are at least four possible ways in which this physical infrastructure could be provided.

The first is through *direct acquisition*; by providing large amounts of funding, new testbeds could be directly purchased. However, the cost of purchasing very large parallel systems could be prohibitively expensive.

A second approach would be to develop *sharing and sandboxing mechanisms* by which "guests" could use the computational resources at other institutions. The sharing mechanisms would need to ensure appropriate priority schemes such that the guests would not pre-empt the compute time of the presumably more important local users. Sandboxing mechanisms to ensure that the guests could not access any local data would also be needed.

A third approach is using a virtual machine such that one computer architecture is made to appear like another through the use of complex software. One disadvantage of early virtual machine systems was as significant performance cost, but more recent work in this area has reduced this penalty. To date however, there is no virtual machine system for large parallel architectures; as such this could be a project worthy of government investment. However, creating parallel architecture from a physical architecture which is not parallel is problematic and it could probably only be done with large performance penalties. However, projects dealing with non-performance related aspects such as fault tolerance, functionality, and correctness would be feasible. One advantage of virtual machine architectures is that they can they run unmodified applications.

A forth approach is developing *simulation platforms*. With realistic timing models, these simulation platforms could even provide realistic performance evaluations. However, simulated systems cannot run unmodified applications as can virtual machines. Another advantage of simulation is that they can provide insight into future system development by exploring trends and simulating systems that do not currently exist.

Virtual Infrastructure

In addition to needing testbeds on which to conduct HEC research, would-be HEC researchers also need access to central clearinghouses of HEC data. Data that would be useful to these researchers include trace data from real HEC applications, synthetic applications that approximate the behavior of these applications, and historical data about failure rates of HEC systems.

Challenges in providing this data are both political and technical. Politically, it may impractical to provide data which may be classified to outside entities. Technical challenges in providing traces or synthetic applications involve the level of detail to provide. As the focus of this workshop is on HEC I/O, one simple answer would be to provide data only about the file system activity of the applications. However, this detail may be insufficient as other aspects of the applications' behaviors may influence their interaction with the file system. For example, a trace of an HEC application run on a machine with a large buffer cache may show less file system activity due to caching than the same application run on a machine with a smaller buffer cache. Additionally, storage and distribution solutions for this clearinghouse data need to be designed and implemented.

Support Growth of I/O Students

Finally, the workshop would like to recognize the vital importance of identifying and supporting students working in the general area of I/O as well as students working more specifically on I/O within HEC. Investment to support the research of these students is worthwhile both because they may provide important research while still in school as well as by cultivating these students such that they may continue to work on HEC I/O following their graduation.

Conclusion

In the near future, sites will deploy supercomputers with tens of thousands of processors routinely, perhaps even hundreds of thousands. Bandwidth needs to storage will go from tens of gigabytes/sec to terabytes/sec. Online storage requirements to support work flows for efficient complex science will begin to approach the exabyte range. The ability to handle a more varied I/O workload ranging seven orders of magnitude in performance characteristics, extremely high metadata activities, and management of trillions of files will be required. Global or virtual enterprise and wide-area sharing of data with flexible and effective security will be required. Current extreme-scale file system deployments already suffer from reliability and availability issues, including recovery times from corruption issues and rebuild times. As these extreme-scale deployments grow larger, these issues will only get worse. It will possibly be unthinkable for a site to run a file system check utility, yet it is almost a given that corruption issues will arise. Recovery times need to be reduced by orders of magnitude, and these types of tools need to be reliable, even though they may rarely be used. The number of storage devices needed in a single coordinated operation could be in the tens to hundreds of thousands, requiring integrity and reliability schemes that are far more scalable than available today. Management of enterprise-class global parallel file/storage systems will become increasingly difficult due to the number of elements involved, which will likely approach 100,000 spinning disks, and widely varying workloads. The challenges of the future are formidable.

The publishing of the document, “HPC File Systems and Scalable I/O: Suggested Research and Development Topics for the Fiscal 2005-2009 Time Frame” and the designation of file systems and I/O as a national focus area beginning in FY06 laid the framework for the HEC-IWG File Systems and I/O Research Guidance Workshop. This workshop helped identify, categorize, and prioritize the needed research in this area of HEC. Using

- the research topics document;
- the workshop output;
- and the Information Storage Industry Consortium (INSIC) roadmap, a roadmap for industry pre-competitive research in the storage area;

the HEC-IWG can embark on a better coordinated government funded research portfolio.

More focused and complete government investment needs to be made in this area of HEC, given its importance and its lack of sufficient funding levels in the past, compared to other elements of HEC. Scalable I/O is perhaps the most overlooked area of HEC research, and given the information generating capabilities being installed and contemplated, it is a mistake to continue to neglect this area of HEC. The HEC-IWG thanks the workshop participants and responders to the questionnaire for providing input into this effort to coordinate government funding in file systems and I/O research.